

CLAIMS

What is claimed is:

1. A method for transforming a datastream comprising the steps of:

a) parsing the datastream into a plurality of work units in a first format; and

5 b) processing each of the plurality of work units by at least one compute node to convert each work unit into a second format.

2. The method of claim 1, wherein the parsing step (a) includes:

10 (a1) providing a plurality of sources, wherein each source is associated with at least one transform;

(a2) instantiating at least one source of the plurality of sources, wherein the at least one instantiated source is associated with the datastream format; and

(a3) utilizing the at least one source to parse the datastream.

15 3. The method of claim 2, wherein the processing step (b) includes:

(b1) loading the at least one transform associated with the at least one instantiated source in the at least one compute node; and

20 (b2) utilizing the at least one transform to convert a work unit of the plurality of work units from the first format to the second format.

4. The method of claim 2 further comprising:

(c) load balancing the plurality of work units.

5. The method of claim 4 wherein the load balancing step includes:
- (c1) generating at least one queue for the plurality of work units; and
 - (c2) distributing each work unit from the at least one queue to the at least one compute nodes in an order.

5

6. The method of claim 5 further comprising:
- (d) returning the plurality of processed work units from the at least one compute node to the at least one source in the order it was distributed.

- 10 7. The method of claim 1, wherein at least one of the plurality of work units is a control work unit that includes control commands for the at least one compute node.

8. The method of claim 7, wherein the processing step (b) includes
- (b1) executing the control commands in the control work unit.

15

9. The method of claim 2, wherein the at least one source is instantiated as a dynamic library.

10. A computer readable medium containing program instructions for transforming a datastream, the program instructions for:

20

- a) parsing the datastream into a plurality of work units in a first format; and
- b) processing each of the plurality of work units by at least one compute node to convert each work unit into a second format.

11. The computer readable medium of claim 10, wherein the parsing instruction (a) includes:

(a1) providing a plurality of sources, wherein each source is associated with at least one transform;

5 (a2) instantiating at least one source of the plurality of sources, wherein the at least one instantiated source is associated with the datastream format; and

a3) utilizing the at least one source to parse the datastream.

10 12. The computer readable medium of claim 11, wherein the processing instruction (b) includes:

(b1) loading the at least one transform associated with the at least one instantiated source in the at least one compute node; and

(b2) utilizing the at least one transform to convert a work unit of the plurality of work units from the first format to the second format.

15 13. The computer readable medium of claim 11 further comprising:

(c) load balancing the plurality of work units.

20 14. The computer readable medium of claim 13, wherein the load balancing instruction includes:

(c1) generating at least one queue for the plurality of work units; and

(c2) distributing each work unit from the at least one queue to the at least one compute nodes in an order.

15. The computer readable medium of claim 14 further comprising:

(d) returning the plurality of processed work units from the at least one compute node to the at least one source in the order it was distributed.

5 16. The computer readable medium of claim 10, wherein at least one of the plurality of work units is a control work unit that includes control commands for the at least one compute node.

10 17. The computer readable medium of claim 16, wherein the processing instruction (b) includes

(b1) executing the control commands in the control work unit.

18. The computer readable medium of claim 11, wherein the at least one source is instantiated as a dynamic library.

15 19. A system for transforming a datastream comprising:
a central component for receiving the datastream in a first format;
a plurality of sources in the central component, wherein each of the plurality of sources is associated with at least one transform; and

20 at least one compute node coupled to the central component,
wherein the central component instantiates at least one source of the plurality of sources that parses the datastream into a plurality of work units in the first format, and

distributes each of the work units to the at least one compute node, wherein the at least one compute node converts each work unit into a second format.

20. The system of claim 19, wherein each of the at least one compute nodes loads
5 the at least one transform as a dynamic library and utilizes the at least one transforms to convert a work unit in the first format to the second format.

21. The system of claim 19, wherein the central component further includes:
a load balancing mechanism coupled to the at least one source for distributing the
10 plurality of work units to the at least one compute node, wherein the load balancing mechanism generates at least one queue for the plurality of work units and dispatches each work unit from the at least one queue to the at least one compute node in an order received from the at least one source.

22. The system of claim 21, wherein the work units processed by the at least one
15 compute node are returned to the at least one source in the order in which the work units were dispatched.

23. The system of claim 19, wherein at least one of the plurality of work units is a
20 control work unit that includes commands for the at least one compute node.

24. The system of claim 23, wherein the at least one compute node processes the control work unit by executing the command.

25. The system of claim 19, wherein the at least one source is instantiated as a dynamic library.